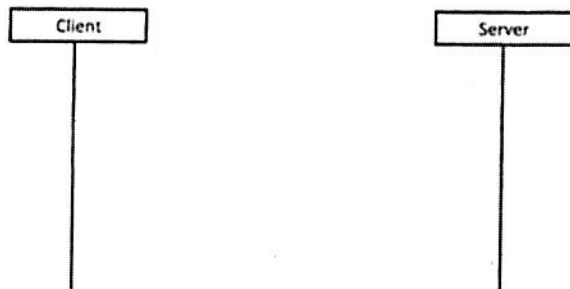


**Aufgabe 1: Transmission Control Protocol**

(5 Punkte)

☐

- a) Nennen und ordnen Sie die grundlegenden Ebenen des TCP/IP Protokoll-Stacks gemäß Vorlesung. Nicht-eindeutige Angaben (z. B. Mehrfachnennungen) werden mit 0 Punkten bewertet. (2 P) ☐
- b) Geben Sie die wesentlichen Schritte des *Three-Way Handshake*-Verfahrens zum Aufbau einer TCP-Verbindung an. Zeichnen Sie dazu die zeitlichen Abläufe mit entsprechender Annotation in das vorgegebene Client-Server-Sequenzdiagramm ein. Nicht-eindeutige Angaben werden mit 0 Punkten bewertet. (3 P) ☐

**Aufgabe 2: HTTP**

(12 Punkte)

☐

- a) Nennen Sie 4 mögliche Request-Methoden bei HTTP und erläutern sie kurz deren Zweck. (4 P) ☐
- Sie je eine idempotente und eine nicht-idempotente Methode. (2 P) ☐
- c) Wie kann man mit Hilfe der GET-Methode Informationen an den Server schicken? Nennen Sie ein konkretes Beispiel. Wann sollte man stattdessen die POST-Methode verwenden? (3 P) ☐
- d) Nennen und erläutern Sie 3 Schwachstellen von HTTP 1.0. (3 P) ☐

**Aufgabe 3: Semantic Web**

(14 Punkte)

☐

- a) Gegeben ist die RDF-Repräsentation *hitchhikers.rdf* (siehe vorletzte Seite der Klausur). Skizzieren Sie die vollständige graphische Darstellung dieses RDF-Dokuments. (11 P) ☐
- b) Das RDF zugrunde liegende Datenmodell besteht aus Tripeln (Subjekt, Prädikat, Objekt). Geben Sie alle Subjekt-Prädikat-Objekt Tripel für das Subjekt *ex:in\_search\_of* an. Sie können die URLs in abgekürzter Schreibweise unter Nutzung der Präfixe *rdf*, *rdfs* und *ex* (wie im RDF Dokument deklariert) angeben. (3 P) ☐

**Aufgabe 4: XML, XPath und JSON**

(8 Punkte)

☐

- a) Nennen Sie 3 Merkmale der Wohlgeformtheit eines XML-Dokuments. Was versteht man unter der Validität eines XML-Dokuments? (2 P) ☐
- b) Gegeben ist die XML-Datei *countries.xml* (letzte Seite der Klausur). Geben Sie jeweils einen XPath-Ausdruck an, der die beschriebenen Elemente exakt referenziert. Der Kontextknoten für alle Ausdrücke ist der Wurzelknoten. (2 P) ☐
- Die Hauptstädte aller Länder als Text
  - Die Währung des Landes mit dem Code "JP" als Text
- c) Geben sie eine gültige JSON-Notation an, welche die Daten der XML-Datei *countries.xml* (letzte Seite der Klausur) repräsentiert. (4 P) ☐

**Aufgabe 5: DTD und XML Schema**

(9 Punkte)

☐

- a) Kreuzen Sie im Folgenden an, ob die Aussage auf DTDs und/oder XML Schema zur Beschreibung von XML-Dokumenten zutrifft. (3 P) ☐

Aussage	DTD	XML Schema
Harmonisiert gut mit Namensräumen	<input type="checkbox"/>	<input type="checkbox"/>
Kann von einem XML-Parser eingelesen werden	<input type="checkbox"/>	<input type="checkbox"/>
Es kann ein bestimmter Namespace für Elemente vorgeschrieben werden	<input type="checkbox"/>	<input type="checkbox"/>
Typen können definiert, abgeleitet und vererbt werden	<input type="checkbox"/>	<input type="checkbox"/>
Die Anzahl von Elementen kann genau festgelegt werden (z.B. genau drei A und drei B Elemente in beliebiger Reihenfolge)	<input type="checkbox"/>	<input type="checkbox"/>
Attributwerte können mittels eines regulären Ausdrucks genauer definiert werden	<input type="checkbox"/>	<input type="checkbox"/>

Jede richtig angekreuzte Aussage wird mit 0,5 Punkten bewertet. Jede falsch oder nicht angekreuzte Aussage wird mit 0 Punkten bewertet.

- b) Es soll ein Katalog für Smartphones in XML definiert werden, der folgende Eigenschaften aufweist:

- Der Wurzelknoten soll Catalog sein und muss mindestens ein Smartphone als Kindknoten enthalten

- Als Kindknoten von Smartphone müssen folgende zwei Elemente in dieser Reihenfolge enthalten sein: Display und Camera

- Jedes Smartphone muss ein Attribut namens Name besitzen, das einen beliebigen Textinhalt haben kann

- Jedes Smartphone muss ein Attribut namens Plattform besitzen, das einen der folgenden drei Werte annehmen kann:

iOS, Android, Windows

- Es kann für Smartphone ein Attribut namens Manufacturer geben, das einen beliebigen Textinhalt haben kann.

Schreiben Sie eine DTD für einen solchen Katalog (Elemente Catalog, Smartphone und ggf. Attribute), sodass obige Eigenschaften exakt erfüllt werden. Die Elemente Display und Camera müssen selbst nicht definiert werden!

- c) Wie könnte ein minimales Instanzdokument dazu aussehen? Ergänzen Sie unten ein XML Dokument, das die Eigenschaften aus Teilaufgabe b) erfüllt und minimal ist, d.h. nur die zwingend notwendigen Elemente und Attribute enthält!

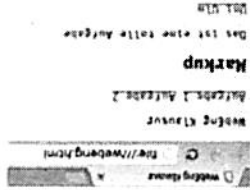
Gebenfalls notwendige Textinhalte können frei gewählt werden.

```
<?xml version="1.0" standalone="no"?>
<IDOCATALOG SYSTEM "catalog.dtd">
```

## Aufgabe 6: HTML und CSS

- a) Vervollständigen Sie das HTML-Markup für die abgebildete Webseite. Achten Sie darauf, semantische HTML5-Elemente zu verwenden. Links müssen nicht auf ein spezielles Ziel verweisen.

(4P)



```
<DOCTYPE html>
<html>
<head>
<title>WebEng Klausur</title>
</head>
</html>
```

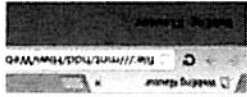
(9 Punkte)

☐
☐
☐
☐

- b) Die obige Webseite soll nun durch CSS so angepasst werden, dass

- der Header grau unterlegt
- die Links zu den einzelnen Aufgaben schwarz
- und der Link im Footer grün

dargestellt wird. Welche Möglichkeiten gibt es, die entsprechenden Elemente auszuwählen? Nennen Sie zwei Möglichkeiten. Geben Sie für eine der Möglichkeiten den CSS-Code für die oben aufgelisteten Anpassungen am HTML-Markup an.



Aufgabe 1 Aufgabe 2

Markup

Das ist eine tolle Aufgabe

WebEng

- c) Wie muss der obige HTML-Code angepasst werden, damit das CSS Stylesheet berücksichtigt wird? Gehen Sie davon aus, dass Sie dieses in der Datei style.css definiert haben.

(1P)

☐

## Aufgabe 8: Serverseitige Programmierung

- a) Wie erfolgt bei CGI die Übergabe von GET-Request-Parametern an den CGI-Prozess? Wie funktioniert die Parameterübergabe bei POST-Requests? (1P)

(1P)

☐

- b) Nennen Sie 2 bekannte Probleme von CGI bei der Erstellung dynamischer Webinhalte. (1P)

(1P)

☐

- c) Wie stehen Java Server Pages (JSPs) im Zusammenhang mit Java Servlets? Zählen Sie kurz stichpunktartig die Schritte bei der Verarbeitung einer JSP-Seite auf. (2P)

(2P)

☐

d) Im Folgenden sehen Sie den Code eines kleinen node.js-Servers. Versuchen Sie zunächst herauszufinden, was dieser Server bewerkstelligen sollte. Bearbeiten Sie dann die folgenden Aufgabenstellungen.

- Beschreiben Sie kurz was dieser Server für eine Aufgabe erfüllen soll
- Im Code befindet sich ein Fehler, der die HTTP-Konformität des Servers bricht. Markieren und verbessern Sie diesen Fehler deutlich im gegebenen Code
- Welches Sicherheitsproblem (unabhängig vom vorher verbesserten Fehler) erkennen Sie in diesem Beispiel? Umreißen Sie einen Lösungsansatz zur Behebung des Problems.

(5 P)

```
var http = require('http');
var url = require('url');
var filesystem = require('fs');

var server = http.createServer(function(request, response) {
  switch(request.method) {
    case "GET":
      handleGet(request, response);
      break;
    default:
      response.writeHead(501, 'Not Implemented');
      response.end();
      break;
  }
});

var handleGet = function(request, response) {
  var path = 'htmlfolder' + url.parse(request.url).pathname;
  filesystem.readFile(path, function(err, data) {
    if (err) {
      response.writeHead(404, 'Not Found');
      response.end();
    } else {
      response.writeHead(204, {
        'Content-Type': 'text/html',
        'Content-Length': data.length
      });
      response.write(data);
      response.end();
    }
  });
};

server.listen(8080);
```

### Aufgabe 9: Clientseitige Programmierung

(10 Punkte)

- a) Wofür steht die Abkürzung AJAX? Erklären Sie, wofür AJAX verwendet wird und wie es funktioniert. (2P)
- b) Nennen Sie jeweils zwei Vor- und Nachteile bei der Verwendung von AJAX. (2P)
- c) Erklären Sie das Konzept eines Cross Site Scripting (XSS-)Angriffs. Welche Maßnahmen kann man ergreifen, um eine Webseite vor XSS-Angriffen zu schützen? (2P)

d) Vervollständigen Sie den folgenden Javascript-Code, der eine Autocomplete-Funktion mit Hilfe von AJAX realisieren soll. (4 P)

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function showHint(str) {
        if (str.length==0) {
          document.getElementById("txtHint").innerHTML="";
          return;
        }
        var xmlhttp=new _____;

        xmlhttp._____ =function() {

          if (xmlhttp._____==4 && xmlhttp._____==_____) {

            document.getElementById("txtHint").innerHTML=xmlhttp._____
          }

        }

        xmlhttp.open("_____", "gethint.php?q="+str, true);

        xmlhttp._____
      }
    </script>
  </head>
  <body>
    <h3>Start typing a name in the input field below:</h3>
    <form action="">
      First name:
      <input type="text" id="txt1" onkeyup="showHint(this.value)" />
    </form>
    <p>Suggestions: <span id="txtHint"></span></p>
  </body>
</html>
```

### hitchhikers.rdf

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:ex="http://example.org/">

  <rdf:Description rdf:about="http://example.org/arthurdent">
    <rdf:type rdf:resource="http://example.org/person"/>
    <rdfs:label xml:lang="en">Arthur Philip Dent</rdfs:label>
    <ex:is_from rdf:resource="http://example.org/earth"/>
    <ex:is_search_of rdf:resource="http://example.org/tea"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/is_from">
    <rdf:type rdf:resource="
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="http://example.org/person"/>
    <rdfs:range rdf:resource="http://example.org/planet"/>
  </rdf:Description>
```

```
<rdf:Description rdf:about="http://example.org/earth">
<rdf:type rdf:resource="http://example.org/planet"/>
<rdf:label xml:lang="en">The Blue Planet</rdf:label>
</rdf:Description>
<rdf:Description rdf:about="http://example.org/tea">
<rdf:type rdf:resource="http://example.org/item"/>
<ex:located-on rdf:resource="http://example.org/shipship"/>
</rdf:Description>
<rdf:Description rdf:about="http://example.org/spacing">
<rdf:type rdf:resource="http://example.org/location"/>
<rdf:label xml:lang="en">Heart of Gold</rdf:label>
</rdf:Description>
<rdf:Description rdf:about="http://example.org/in-reach-of">
<rdf:type rdf:resource="http://www.w3.org/1999/02-22-rdf-syntax-ns#Property"/>
<ex:domain rdf:resource="http://example.org/person"/>
<rdf:range rdf:resource="http://example.org/item"/>
</rdf:Description>
</rdf:RDF>
<?xml version="1.0" encoding="UTF-8"?>
<countries>
<country>
<name countryCode="DE">Germany</name>
<currency>Euro</currency>
<capital>Berlin</capital>
</country>
<country>
<name countryCode="US">United States</name>
<currency>United States dollar</currency>
<capital>Washington, D.C.</capital>
</country>
<country>
<name countryCode="JP">Japan</name>
<currency>Yen</currency>
<capital>Tokyo</capital>
</country>
</countries>
```

## countries.xml

## Aufgabe 1: Identifikation von Ressourcen

(5 Punkte)

(1 P)

a) Uniforme Identifikation

Veranschaulichen Sie durch Pfeile (A → B: B ist eine Teilmenge von A) die Relationen zwischen Uniform Resource Name (URN), Uniform Resource Identifier (URI) und Uniform Resource Locator (URL) gemäß des "Classical Views" aus RFC 3305.

Uniform Resource Name (URN)

Uniform Resource Identifier (URI)

Uniform Resource Locator (URL)

b) Uniform Resource Identifier (URI)

(2,5 P)

Zerlegen Sie die gegebenen URI in ihre einzelne Bestandteile gemäß RFC 3986:

foo://example.com:8042/select/port?name=value#eof

authority:

fragment:

path:

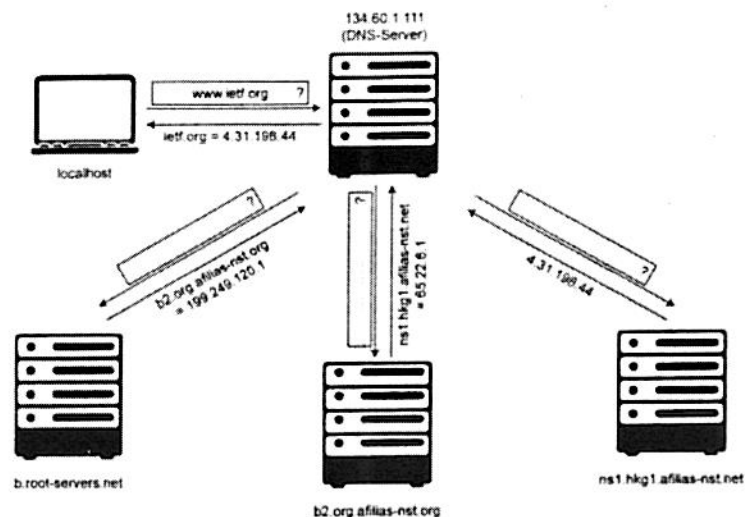
query:

scheme:

## c) Domain Name System (DNS)

(1,5 P)

Das Schaubild zeigt die Anfrage eines Clients an einen DNS-Server. Ergänzen Sie die Kommunikation zwischen dem DNS-Server und den Root- und Nameservern. Tragen Sie dazu den Teil der Adresse `www.ietf.org`, der beim angefragten Server aufgelöst werden soll, in den jeweiligen Kasten im Schaubild ein.



## Aufgabe 2: HTTP

(15 Punkte)

- a) Nennen Sie 4 der möglichen Request-Methoden bei HTTP und erläutern sie kurz deren Zweck. (4 P)
- b) Nennen und erläutern sie 3 technologische Nachteile von HTTP 1.0. (3 P)
- c) Nennen sie die 2 Authentisierungsverfahren, die sie bei HTTP kennengelernt haben. (1 P)
- d) Wie unterscheiden sich diese beiden Verfahren? (4 P)
- e) Bei welchem dieser Verfahren wird eine sogenannte "Nonce" verwendet? Erklären Sie was eine "Nonce" ist und wofür sie verwendet wird. (2 P)
- f) Wofür ist die Angabe von MIME-Ressourcentypen bei HTTP wichtig? Nennen Sie 2 Funktionen heutiger Browser die durch Ressourcentypen ermöglicht werden. (1 P)

## Aufgabe 3: Semantic Web

(14 Punkte)

- a) Gegeben ist die RDF-Repräsentation `gameofthrones.rdf` (siehe vorletzte Seite der Klausur). Skizzieren Sie die vollständige graphische Darstellung dieses RDF-Dokuments. Sie können die URLs hierbei in abgekürzter Schreibweise unter Nutzung der Präfixe `rdf`, `rdfs` und `ex` (wie im RDF Dokument deklariert) angeben, zB. `ex:jonsnow`, `rdfs:label`, und `rdf:type`. (12 P)
- b) Das RDF zugrunde liegende Datenmodell besteht aus Tripeln (Subjekt, Prädikat, Objekt). Geben Sie alle vollständigen Subjekt-Prädikat-Objekt Tripel für das Subjekt `direwolf` an. (2 P)

## Aufgabe 4: XML, XPath und JSON

(10 Punkte)

- a) Gegebene ist die folgende Liste von XPATH-Ausdrücken und die XML-Datei `artists.xml` (letzte Seite der Klausur).

(6 P)

- (1) `//name[../@genre="Rock"]`
- (2) `//died[@aged="27"]/..//name`
- (3) `/artists/soloartist[nickname = "2Pac"]/name`
- (4) `//name[../@genre="Rock"and ../died > 1975]`
- (5) `/artists/band[nickname = "The King"]/name`
- (6) `//name[.="The Beatles"]/..//name`

Kreuzen Sie im Folgenden an, ob die angegebenen Knoten Teil der Ergebnisliste der jeweiligen XPATH-Ausdrücke sind.

Ausdruck	The Beatles	John Lennon	Elvis Presley	Jimi Hendrix	Tupac Shakur
(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(4)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(5)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(6)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Jede richtig angekreuzte Zeile wird mit 1 Punkt bewertet. Jede falsch oder nicht angekreuzte Zeile wird mit 0 Punkten bewertet.

- b) Geben sie eine gültige JSON-Notation an, welche die Daten der Artisten **The Beatles** und **Elvis Presley** aus der XML-Datei `artists.xml` (letzte Seite der Klausur) repräsentiert.

(4 P)

## Aufgabe 5: DTD und XML Schema

(8 Punkte)

- a) Kreuzen Sie im Folgenden an, ob die Aussage auf DTDs, auf XML Schema, oder auf Beides zur Beschreibung von XML-Dokumenten zutrifft. (3 P)

Aussage	XML Schema	DTD
Attributwerte können mittels eines regulären Ausdrucks genauer definiert werden	<input type="checkbox"/>	<input type="checkbox"/>
Attribute können als optional definiert werden	<input type="checkbox"/>	<input type="checkbox"/>
Ist eine XML Anwendung (kein eigener Parser notwendig)	<input type="checkbox"/>	<input type="checkbox"/>
Es kann ein Element definiert werden, das sowohl Text als auch andere Elemente enthält	<input type="checkbox"/>	<input type="checkbox"/>
Es können Entities definiert werden	<input type="checkbox"/>	<input type="checkbox"/>
Inhalte können typisiert werden	<input type="checkbox"/>	<input type="checkbox"/>

Jede richtig angekreuzte Aussage wird mit 0,5 Punkten bewertet. Jede falsch oder nicht angekreuzte Aussage wird mit 0 Punkten bewertet.

b) Gegeben sei folgendes XML Schema:

(5P)



```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```

  <xs:element name="Smartwatches">
    <xs:sequence>
      <xs:element name="Smartwatch" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType base="Smartwatch" type="displaytype">
          <xs:attribute name="dimensionX" type="integer" use="required"/>
          <xs:attribute name="dimensionY" type="integer" use="required"/>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="round"/>
              <xs:enumeration value="square"/>
            </xs:restriction>
          </xs:simpleType>
          <xs:attribute name="formfactor" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:element>

  <xs:element name="Smartwatch">
    <xs:sequence>
      <xs:element name="display" type="displaytype"/>
      <xs:element name="sensors" minOccurs="0" maxOccurs="1"/>
      <xs:element name="battery" type="batterytype"/>
      <xs:element name="watchband" type="watchbandtype"/>
    </xs:sequence>
    <xs:attribute name="Name" use="required"/>
  </xs:complexType>
</xs:element>

  <xs:element>
    <xs:sequence>
      <xs:element>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="GearS">
              <xs:display dimensionX="220" dimensionY="240" />
              <xs:sensors>
                <xs:accelerometer>
                  <xs:sensor>Gyroscope</sensor>
                </xs:sensors>
                <xs:battery capacity="3500" />
                <xs:watchband length="23.5">Steel</watchband>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:element>

  <xs:element name="Smartwatch Name="LGWatch">
    <xs:battery capacity="1100" />
    <xs:sensors>
      <xs:accelerometer>
        <xs:sensor>Accelerometer</sensor>
      </xs:sensors>
    </xs:complexType>
  </xs:element>

  <xs:element name="Smartwatch">
    <xs:sequence>
      <xs:display dimensionX="180" dimensionY="220" />
      <xs:sensors>
        <xs:gyroscope</sensor>
      </xs:sensors>
      <xs:watchband length="23.5" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

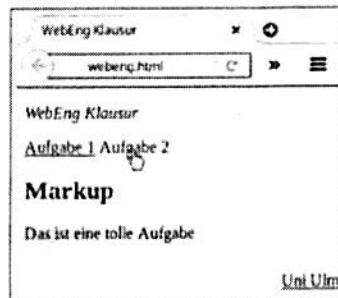
In folgendem Instanzdokument haben sich bezüglich des obigen Schemas 5 Fehler eingeschli-

chen. Markieren Sie diese und geben sie den Grund für den Fehler an.

**Aufgabe 6: HTML und CSS**

(10 Punkte)

- a) Was kann innerhalb des HTML `<head>`-Elements spezifiziert werden? Nennen Sie 3 Beispiele und erklären Sie diese jeweils kurz. (3P)
- b) Nennen und erklären Sie eine Möglichkeit, um Benutzern, die eine Webseite mit einem Smartphone besuchen, eine entsprechend angepasste Version der Webseite auszuliefern. Schreiben Sie außerdem ein Beispiel, wie dies in Code angegeben wird. (3P)
- c) Erweitern Sie den vorgegebenen HTML-Markup mit CSS-Regeln, um die folgenden Gestaltungseigenschaften zu erfüllen: (4P)
- Die Überschrift soll kursiv dargestellt werden
  - Die beiden Links sollen im Normalzustand unterstrichen sein
  - Bewegt sich der Cursor auf einen Link, so soll dieser nicht mehr unterstrichen sein
  - Der footer soll unten rechts dargestellt werden.



```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>WebEng Klausur</title>

    <style>

  </style>
</head>
<body>
  <header>
    <p>WebEng Klausur</p>
  </header>
  <nav>
    <a href="#">Aufgabe 1</a>
    <a href="#">Aufgabe 2</a>
  </nav>
  <article>
    <h1>Markup</h1>
    <p>Das ist eine tolle Aufgabe</p>
  </article>
</body>
</html>
```

**Aufgabe 7: Web-Architekturen**

(9 Punkte)

- a) Wofür werden Proxies verwendet? Nennen und beschreiben Sie drei Aufgaben. (3P)
- b) Nennen Sie zwei Strategien zur Server-Auswahl beim Web Switching. (1P)
- c) Was ist ein Cache Miss? Beschreiben Sie kurz das Konzept, sowie was nach einem solchen Vorfall passiert. (1P)
- d) Beschreiben Sie allgemein das Prinzip der Cache-Validierung, sowie einen Vorteil dieses Prinzips. Nennen Sie einen HTTP-Header der für diesen Zweck vom Cache verwendet wird. (3P)
- e) MPMs (Multi-Processing Modules) regeln bei Apache 2 die Zuordnung von eingehenden Verbindungen zu mehreren verarbeitenden Prozessen/Threads. Beschreiben Sie kurz einen Vorteil dieses Ansatzes im Vergleich zu strikt sequentieller Verarbeitung. (1P)

**Aufgabe 8: Serverseitige Programmierung**

(9 Punkte)

- a) Nennen Sie 4 Aufgaben die ein Java Servlet Container (auch bekannt als Servlet Engine) erfüllt. (2P)
- b) Nennen Sie 2 Nachteile von Servlets die letztendlich zur Entwicklung von JSP geführt haben. (1P)
- c) Im Folgenden sehen Sie den Code eines kleinen Python-Servers. Versuchen Sie zunächst herauszufinden, was dieser Server bewerkstelligt. Sie können hierzu einige Annahmen machen:
- Es existiert eine valide SQLite Datenbankdatei namens `database.db` im Verzeichnis des Servers. Die Datenbank beinhaltet eine Tabelle namens `superheroes`, welche aus 4 Spalten (`name`, `realname`, `universe`, `powers`) besteht.
  - Clients wissen, dass direkt der Pfad `/superheroes/` aufzurufen ist um den Server zu verwenden.

Bearbeiten Sie nun die folgenden Aufgabenstellungen:

- Beschreiben Sie detailliert die Funktionalitäten dieses Servers. (3 Punkte)
- Welches offensichtliche Sicherheitsproblem erkennen Sie in diesem Codebeispiel? Beschreiben Sie diese Sicherheitslücke kurz, mit Referenz auf relevante Zeilen im Code. (1 Punkt)
- Umreißen Sie zudem einen Lösungsansatz zur Behebung des Problems. Hierzu ist zumindest eine Beschreibung Ihrer Vorgehensweise bei der Behebung der Sicherheitslücke erforderlich. Es ist jedoch keine Demonstration korrekten Pythoncodes nötig. (2 Punkte)

(6P)



- c) Im Folgenden sehen Sie den Code eines kleinen Python-Servers. Versuchen Sie zunächst herauszufinden, was dieser Server bewerkstelligt. Sie können hierzu einige Annahmen machen:
- Es existiert eine valide SQLite Datenbankdatei namens `database.db` im Verzeichnis des Servers. Die Datenbank beinhaltet eine Tabelle namens `superheroes`, welche aus 4 Spalten (`name`, `realname`, `universe`, `powers`) besteht.
  - Clients wissen, dass direkt der Pfad `/superheroes/` aufrufen ist um den Server zu verwenden.
- Bearbeiten Sie nun die folgenden Aufgabenstellungen:
- Beschreiben Sie detailliert die Funktionalitäten dieses Servers. (3 Punkte)
  - Welches offensichtliche Sicherheitsproblem erkennen Sie in diesem Codebeispiel? Beschreiben Sie diese Sicherheitslücke kurz, mit Referenz auf relevante Zeilen im Code. (1 Punkt)
  - Umreißen Sie zudem einen Lösungsansatz zur Behebung des Problems. Hierzu ist zumindest eine Beschreibung ihrer Vorgehensweise bei der Behebung der Sicherheitslücke erforderlich. Es ist jedoch keine Demonstration korrekten Pythoncodes nötig. (2 Punkte)

(6P)

```
1 from flask import Flask
2 from flask import jsonify
3 import sqlite3
4 app = Flask(__name__)
5
6 def query_db(query):
7     db = sqlite3.connect('database.db')
8     cur = db.execute(query)
9     data = cur.fetchall()
10    cur.close()
11    db.close()
12    columns = [col[0].lower() for col in cur.description]
13    return map(lambda d:dict(zip(columns, d)), data)
14
15 @app.route('/superheroes/', defaults={'hero_name': None, methods=['GET']})
16 @app.route('/superheroes/<hero_name>', methods=['GET'])
17 def get_superheroes(hero_name):
18     query = 'SELECT * FROM superheroes'
19     if hero_name is not None:
20         query += ' WHERE name = \'' + hero_name + '\''
21     db_result = query_db(query)
22     tmp_res_object = {'heroes': []}
23     for hero in db_result:
24         tmp_res_object['heroes'].append(
25             {'name': hero['name'],
26              'realname': hero['realname'],
27              'universe': hero['universe'],
28              'powers': hero['powers']}
29         )
30     res = jsonify(tmp_res_object)
31     res.status_code = 200
32     return res
33
34 if __name__ == '__main__':
35     app.run()
```

17/24

## Aufgabe 9: Clientseitige Programmierung

- a) Wofür steht AJAX? Wie unterscheidet sich die Interaktion zwischen Client und Server bei einer AJAX-basierter Anwendung von einer klassischen Web-Anwendung? Skizzieren Sie den Interaktionsablauf in beiden Fällen und erklären Sie das Verhalten. (4P)
- b) Was besagt die Same Origin Policy? Welchen Zweck hat sie? (2P)
- c) Nennen Sie außer der Same Origin Policy 2 weitere Beispiele für Einschränkungen der JavaScript-Funktionalität durch Browser und warum diese sinnvoll sind. (2P)
- d) Nutzereingaben in Formulare auf Webseiten lassen sich auch clientseitig mit JavaScript auf Korrektheit prüfen. Nennen Sie einen Vorteil solcher clientseitigen Überprüfungen. Erklären Sie außerdem, warum Eingabeüberprüfungen trotzdem immer nochmals vom Server durchgeführt werden sollten. (2P)

## gameofthrones.rdt

```
<rdt:RDF
  xmlns:rdt="http://www.w3.org/1999/02/22-rdt-syntax-ns#"
  xmlns:ex="http://www.w3.org/2000/01/rdt-schema#"
  xml:base="http://example.org/">
  <rdt:Description rdt:about="http://example.org/johnsnow">
    <rdt:type rdt:resource="http://example.org/person"/>
    <rdts:label xml:lang="en">Jon Snow</rdts:label>
    <ex:part_of rdt:resource="http://example.org/stark"/>
    <ex:has_pet rdt:resource="http://example.org/direwolf"/>
    <ex:works_for rdt:resource="militaryorder"/>
  </rdt:Description>
  <rdt:Description rdt:about="http://example.org/part_of">
    <rdt:type rdt:resource="http://example.org/part_of">
    <rdts:domain rdt:resource="http://example.org/person"/>
    <rdts:range rdt:resource="http://example.org/house"/>
  </rdt:Description>
  <rdt:Description rdt:about="http://example.org/direwolf">
    <rdt:type rdt:resource="http://example.org/animal"/>
    <rdts:label xml:lang="en">Ghost</rdts:label>
  </rdt:Description>
  <rdt:Description rdt:about="http://example.org/militaryorder">
    <rdt:type rdt:resource="http://example.org/organization"/>
    <rdts:label xml:lang="en">Night's Watch</rdts:label>
  </rdt:Description>
  <rdt:Description rdt:about="http://example.org/stark">
    <rdt:type rdt:resource="http://example.org/house"/>
  </rdt:Description>
  <rdts:domain rdt:resource="http://example.org/person"/>
  <rdts:range rdt:resource="http://example.org/animal"/>
  </rdt:Description>
```



```

<rdf:Description rdf:about="http://example.org/direwolf">
<rdf:type rdf:resource="http://example.org/animal"/>
<rdflabel xml:lang="en">Ghost</rdflabel>
</rdf:Description>

<rdf:Description rdf:about="http://example.org/militaryorder">
<rdf:type rdf:resource="http://example.org/organisation"/>
<rdflabel xml:lang="en">Night's Watch</rdflabel>
</rdf:Description>

<rdf:Description rdf:about="http://example.org/stark">
<rdf:type rdf:resource="http://example.org/house"/>
</rdf:Description>

<rdf:Description rdf:about="http://example.org/has_pet">
<rdf:type rdf:resource="
"http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
<rdflabel domain rdf:resource="http://example.org/person"/>
<rdflabel range rdf:resource="http://example.org/animal"/>
</rdf:Description>

<rdf:Description rdf:about="http://example.org/works_for">
<rdf:type rdf:resource="
"http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
<rdflabel domain rdf:resource="http://example.org/person"/>
<rdflabel range rdf:resource="http://example.org/organisation"/>
</rdf:Description>

</rdf:RDF>

```

### artists.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<artists>
  <band genre="Rock">
    <name>The Beatles</name>
    <members>
      <name>John Lennon</name>
      <name>Paul McCartney</name>
      <name>George Harrison</name>
      <name>Ringo Starr</name>
    </members>
  </band>
  <soloartist genre="Rock">
    <name>Elvis Presley</name>
    <nickname>The King</nickname>
    <died aged="45">1977</died>
  </soloartist>
  <soloartist genre="Rock">
    <name>Jimi Hendrix</name>
    <died aged="27">1970</died>
  </soloartist>
  <soloartist genre="Hip Hop">
    <name>Tupac Shakur</name>
    <nickname>2Pac</nickname>
    <died aged="25">1996</died>
  </soloartist>
</artists>

```

### Aufgabe 1: Identifikation von Ressourcen

(7 Punkte)

☐

- a) Eine URI besteht typischerweise aus mehreren Teilen. Geben Sie eine geeignete Beispiel URI an und markieren und benennen Sie drei der Teile. (4 P)

☐

Beispiel:

```

foo://example.com:8042/select/port?name=value#eot
authority  example.com:8042
fragment   eot
path       /select/port
query      name=value
scheme     foo

```

Benotung: 1 Punkt URI, ein Punkt pro markiertem und benannten Teil

- b) Zusätzlich zu URIs haben Sie in der Vorlesung URNs und URLs kennen gelernt. Beschreiben Sie jeweils kurz den Zweck dieser drei Identifizierungskonzepte. (3 P)

☐

URI: Identifizierung einer Ressource in irgendeiner Form

URN: Identifizierung einer Ressource über einen Namen

URL: Identifizierung einer Ressource über eine Adresse

jeweils 1 Punkt pro Erklärung

---

**Aufgabe 3: XML**

(23 Punkte)

- a) Nennen Sie zwei Funktionen die XML Namensräume erfüllen.

(2 P)

Auflösen von Namenskonflikten, Zusammenfassung von zusammengehörigem Vokabular

- b) Ordnen Sie jeweils zu, welche der folgenden Eigenschaften DTDs und XML Schema besitzen. Jede Eigenschaft muss eindeutig einer Seite zugeordnet werden. Richtig genannte Eigenschaften geben 1 Punkt, falsch zugeteilte 0.5 Punkte Abzug. Insgesamt ergibt die Aufgabe mindestens 0 Punkte.

(4 P)

- Einfache und komplexe Datentypen
- Namensraumsensitive Element- und Attributdeklarationen
- Ableitung und Vererbung von Typen
- Definition von Entities

DTD	XML Schema
- Definition von Entities	- Einfache und komplexe Datentypen - Ableitung und Vererbung von Typen - Namensraumsensitive Element- und Attributdeklarationen

Quelle: Kapitel3, Seite 175 Punkteverteilung: siehe Aufgabenbeschreibung

- c) Erklären Sie den konzeptionellen Unterschied zwischen SAX- und DOM-Parsern für XML. Geben Sie jeweils ein Beispielszenario an, für das Sie die Technologie verwenden würden. (5 P)

1.5+1.5 Konzepte, 1+1 Beispiele Beispielszenario. Anwendungsbeispiele sollten auf die Unterschiede der Technologien eingehen, insbesondere in Hinsicht auf große XML Dokumente (DOM komplettes Modell im Speicher und daher gut zu navigieren, aber speicherintensiv; SAX ereignisorientiert und hält nur einen Teil im Speicher)

- d) Erklären Sie die Definition des <tr> Elements laut seiner im folgenden gegebenen HTML4 DTD Definition

&lt;!ELEMENT TR - o (TH|TD)+ &gt;.

(4 P)

- Kinder können TH, TD Elemente sein
- Mindestens ein Kind
- Öffnendes Tag ist verpflichtend
- Schließendes Tag ist optional

Pro genannter Eigenschaft 1 Punkt

- e) Betrachten Sie die an die Klausur angehängten Dokumente nodes.xml und graph.xslt. Geben Sie das von der Transformation erzeugte Dokument an. Zeilenumbrüche können Sie beliebig wählen, Leerzeichen zwischen Worten sind ebenfalls nicht ausschlaggebend. Hinweis: &gt; wird zu dem Operator '>' ausgewertet. (8 P)

```

<html>
  <body>Hogwarts<br />
    <ul>
      <li>Britannia (0.8)</li>
      <li>Paragaphus (0.7)</li>
    </ul>
    Red London<br />
    <ul></ul>
    Britannia<br />
    <ul>
      <li>Ulm (0.9)</li>
    </ul>
    Paragaphus<br />
    <ul>
      <li>Ulm (1.5)</li>
    </ul>
    Ulm<br />
    <ul>
      <li>Hogwarte (0.0)</li>
    </ul>
  </body>
</html>

```

Bepunktung:

- Bis zu 2 Punkte für die generell richtige HTML Struktur, also Abfolge von html, body, br, ul, li. Dabei sind 0.5 Punkte für die leeren Listen reserviert.
- 2 Punkt für die richtigen Hauptelemente (Hogwarts, Red London, Britannia, Paragaphus, Ulm)
- 2 Punkte für prinzipiell richtige Listenformate (also Name plus Gewicht in Klammern)
- Bis zu 2 Punkte für den richtigen Listencontent, also wirklich nur Nodes mit passendem Gewicht.

# Aufgabe 4: Javascript und AJAX

a) In einer Webanwendung sollen Bücher und deren Preise verwaltet werden. Dafür soll eine HTML-Tabelleansicht wie folgend abgebildet über Javascript dynamisch generiert werden. Ebenfalls angegeben sind die übermittelten JSON Daten.

Cover	Price	Link
	7.99	Link
	32	Link
	12	Link

```

{
  "books": [
    {
      "cover": "/img/schwabl.jpg",
      "price": 7.99,
      "link": "some.domain/book/XKCD"
    },
    {
      "cover": "/img/ruby.jpg",
      "price": 32,
      "link": "some.domain/book/XFCE"
    },
    {
      "cover": "/img/moors.jpg",
      "price": 32,
      "link": "some.domain/book/SMBC"
    }
  ]
}
    
```

Vervollständigen Sie den auf der folgenden Seite abgedruckten Javascript Code. Sie können dabei davon ausgehen, dass der Körper der Tabelle mit dem Attribut id='data' versehen ist.

☐

(13P)

☐

(14Punkte)

```

function load(){
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "http://127.0.0.1/webeng/data-min.json");
    xhr.onreadystatechange = function(){
        if(xhr.status == 200 && xhr.readyState == 4){
            //Antwort umwandeln
            var dataObj = JSON.parse(xhr.responseText);
            var tbody = document.getElementById("data");
        }
    }
}
    
```

```

//Einzelne Zellen zusammenbauen
var coverData = "<td<img src='"+ dataObj.books[i].cover + "'></td>";
var priceData = "<td>" + dataObj.books[i].price + "</td>";
var linkData = "<td>a href='"+ dataObj.books[i].link + "'>Link</a>";
//Neue Zeile zusammenbauen und einfügen
tbody.innerHTML += "<tr>" + coverData + priceData + linkData + "</tr>";
    
```

```

}
}
}
xhr.send();
}
    
```

1 Punkt pro richtig eingegebenem freien Feld; JSON.parse ergibt insgesamt nur einen Punkt

b) HTML5 führt unter anderem Browser History Management ein. Welche Bedeutung hat dies für AJAX Anwendungen? (1P)

☐

AJAX Anwendungen können damit einige 'klassische Nachteile' ausgleichen, indem sie passende History Einträge für Zwischenzustände der Webanwendung anlegen.

**Aufgabe 5: HTML und CSS**

(19 Punkte)

- a) Erklären Sie kurz den Unterschied zwischen den beiden folgenden Codefragmenten (2 P)

```
<div class="content">...</div>           <div id="content">...</div>
```

Class: es gibt mehrere gleichwertige Divs, id. eindeutiger Bezeichner

- b) Erklären Sie den Begriff Responsive Design. (1 P)

Website ist so aufgebaut, dass sie auf unterschiedlichen Displaygrößen sinnvoll angezeigt wird.  
 • Hauptsächlich durch CSS media queries ermöglicht.

- c) Klassifizieren Sie die Tags im folgenden Dokument eindeutig in Semantic und Presentational Markup. Richtige Klassifizierung geben 1 Punkt, falsche 0.5 Punkte Abzug, fehlende Einträge ergeben 0 Punkte. Insgesamt ergibt die Aufgabe mindestens 0 Punkte. (8 P)

```
<titel>Teilnehmer</titel>
<block>Folgende Personen haben die <vorlesung>Web
Engineering</vorlesung> Vorlesung besucht</block>
<liste>
<eintrag><person>Hermann Jansen</person></eintrag>
<eintrag><person>Liaris Oswell</person></eintrag>
<eintrag><person>Pascal Brunner</person></eintrag>
<eintrag><person>Birte Glimm</person></eintrag>
</liste>
<block><fett>Die Klausur findet am
<datum>21.02.17</datum> statt</fett></block>
```

Semantic Markup	Presentational Markup
vorlesung	titel
person	liste
datum	eintrag
	block
	fett

Vgl. Kapitel 2, Folie 26

- d) Betrachten Sie das folgende HTML Dokument

```
<!doctype html>
<html>
  <head></head>
  <body>
    <div class="header"> (1)
      <h1>Site Header</h1>
      <div>Subdiv</div>
    </div>
    <div class="content"> (1)
      <h1>Content</h1> (2)
      <table>
        <thead>
          <tr>
            <td>First</td>
            <td>Second</td>
          </tr>
        </thead>
        <tbody>
          <tr class="select"> (3)
            <td>Foo</td> (3)
            <td>Bar</td> (3)
          </tr>
        </tbody>
      </table>
      <div class="select">Baz</div>
    </div>
  </body>
</html>
```

Gegeben sind des weiteren die folgenden CSS Selektoren. In den Zeilen mit öffnenden Tags, die von einem der folgenden Selektoren selektiert werden, geben Sie (am Anfang oder Ende der Zeile) die Nummer (1, 2 oder 3) des jeweiligen Selektors an. (6 P)

- body > div
- .content h1

Pro Selektor 2 Punkte

- e) Welchen Vorteil bietet die Verwendung der neuen HTML5 Elemente wie <footer>, <article>, etc. im Gegensatz zu <div>? (2 P)

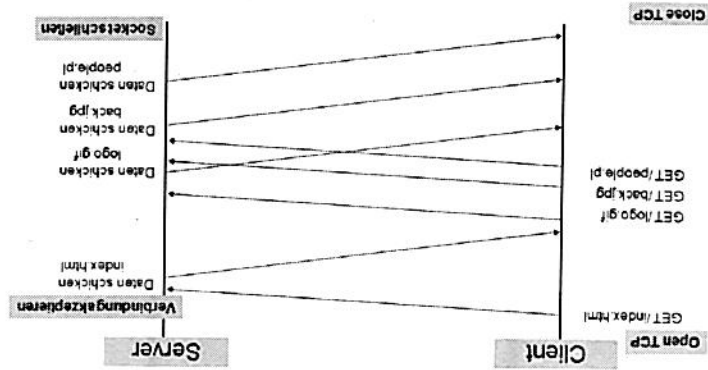
Klar definierte Semantik, erlaubt es, Webseiten besser von Algorithmen verarbeiten zu lassen

## Aufgabe 6: HTTP

a) Erklären Sie anhand der folgenden Grafik HTTP/1.1 Pipelining. Warum wird Pipelining hier nicht für alle Ressourcen eingesetzt?

(3P)

(12 Punkte)



Pipelining: Zu sehen hier an logo.gif, back.jpg und people.pl. Es werden mehrere Anfragen nach Ressourcen nacheinander oder gleichzeitig abgeschickt und nicht erst gewartet, dass eine Ressource fertig übertragen wurde. Auf index.html muss hingegen vollständig (oder teilweise vollständig) gewartet werden, da erst dadurch klar wird, welche weiteren Ressourcen vom Server angefragt werden müssen.

2 Punkte für die Erklärung, 1 Punkt für die Überlegung.

b) HTTP Status Codes werden bereits anhand der ersten Ziffer des dreistelligen Codes grob in Kategorien eingeteilt. Geben Sie für drei der unten stehenden Einträge jeweils an, welche Kategorie von Nachrichten die Codes beschreiben:

(3P)

- 1xx: Informational
- 2xx: Success
- 3xx: Redirection
- 4xx: Client errors
- 5xx: Server errors

c) Der Wechsel von HTTP/1.1 auf HTTP/2.0 bedeutet insbesondere einen Wechsel von einem ASCII Protokoll auf ein Binärprotokoll. Nennen Sie einen Vor- und einen Nachteil dieses Wechsels.

Vorteil: Kompaktere Darstellung, weniger Traffic. Nachteil: Nicht mehr menschlesbar, je 1 Punkt

d) Wählen Sie aus dem folgenden GET Request vier unterschiedliche, unterstrichene Header aus und erklären Sie die Bedeutung des gesetzten Wertes.

(4P)

```

GET /pundaeomontium.png HTTP/1.1
Host: ww.unl-ulm.de
User-Agent: Java/1.6.0.26
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://ww.unl-ulm.de/en/
Cookie: mind=blown
DNT: 1
Connection: keep-alive
If-Modified-Since: Mon, 05 Sep 2016 15:36:03 GMT
If-None-Match: "d9c1-53bc4701513fa"
Cache-Control: max-age=0

```

Einige Möglichkeiten, jeweils 1 Punkt pro Erklärung

- Accept: Welche Formate als Antwort akzeptiert werden, in diesem Fall alle
- Referer: Von wo der Client herkam
- Cookie: Gesetzte Cookies für die Domäne
- Connection: TCP Verbindung soll weiter genutzt werden
- If-None-Match: Caching Kontrolle, wenn das etag nicht übereinstimmt, soll neu gesendet werden
- Cache-Control: Caching Angaben, hier, dass die Resource immer aktuell (maximales Alter 0 s) sein muss

**Aufgabe 7: Ruby on Rails**

(5 Punkte)

☐

Gegeben ist folgende `routes.rb` einer Ruby on Rails Anwendung.

```
Rails.application.routes.draw do
  resources :foo

  resources :bar, only: [:index, :show]
end
```

- a) Geben Sie jeweils an, welche der HTTP Methoden GET, POST, PUT, DELETE auf `/foo/`, `/foo/$id`, `/bar` und `/bar/$id` anwendbar sind, wobei `$id` eine beliebige, im jeweiligen Kontext gültige ID sein soll. (4P)

☐

`/foo:` GET, POST

`/foo/$id:` GET, PUT, DELETE

`/bar:` GET

`/bar/$id:` GET

1 Punkt pro richtiger Zeile

- b) Was versteht man unter *Convention over Configuration* wie Sie es zum Beispiel im Rahmen von Ruby on Rails kennengelernt haben? (1P)

☐

Anstelle von vielen Konfigurationsdateien erfolgt die Zuordnung verschiedener Dinge durch Namenskonventionen, also anstelle ein Objekt zu einer Tabelle mittels Konfiguration zuzuordnen, sorgt der gleiche Name für die Verbindung.

**Aufgabe 8: Web Architekturen**

(11 Punkte)

☐

- a) Erklären Sie den Begriff Domain Sharding und erklären Sie, wozu es eingesetzt wird. (2P)

☐

1 Punkt Erklärung Begriff, 1 Punkt Erklärung Zweck. Domain Sharding ist das Prinzip, Content über mehrere Subdomains zu verteilen. Damit können Ressourcen von mehreren Domains geladen werden, was einige Mechanismen in Hinsicht auf Verbindungsbeschränkungen umgeht und so einen schnelleren Seitenaufbau ermöglicht.

- b) Nennen Sie zwei Vorteile, die die Benutzung eines Content Delivery Networks bietet. (2P)

☐

je 1 Punkt, bessere geographische Nähe zum Benutzer, besseres Management von temporären Belastungsspeaks und ggf. weitere

- c) Welcher Faktor (von Webseiten) liegt dem Google PageRank Algorithmus zu Grunde, d.h. welcher Faktor bestimmt die Relevanz einer Webseite? (1P)

☐

Faktor: Zahl der Verlinkungen durch andere Webseiten

- d) Erklären Sie im Zusammenhang mit Suchmaschinen den Begriff Long Tail Answer. (1P)

☐

Long Tail: Lösung steckt nicht unter den hoch gerankten Seiten, sondern im Long Tail und wird damit von der Suchmaschine weit hinten angezeigt.

- e) Erklären Sie die Schlüsselworte *no-store*, *no-cache*, *private*, *public* und *max-age* im Zusammenhang mit Caching. (5P)

☐

1 Punkt pro Erklärung

- *no-store*: Nichts des Requests oder Response darf gecached werden.
- *no-cache*: Caches müssen einen Request an den Ursprungsserver stellen, um die gecachte Resource vor der nächsten Nutzung zu validieren
- *private*: Response ist für einen bestimmten Nutzer und darf nicht in gemeinsam genutzten Caches gespeichert werden
- *public*: Response kann von beliebigen Caches gecached werden
- *max-age*: Gibt die maximale Zeit in Sekunden an, für die die Resource als aktuell betrachtet werden kann.



## hitichhikers.rdf

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:ex="http://example.org/">
  <rdf:Description rdfs:about="http://example.org/arthurburdett">
    <rdf:type rdfs:resource="http://example.org/person"/>
    <rdfs:label xml:lang="en">Arthur Philip Dent</rdfs:label>
    <ex:is_from rdfs:resource="http://example.org/earth"/>
    <ex:in_search_of rdfs:resource="http://example.org/tea"/>
  </rdf:Description>
  <rdf:Description rdfs:about="http://example.org/as_from">
    <rdf:type rdfs:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdfs:resource="http://example.org/person"/>
    <rdfs:range rdfs:resource="http://example.org/planet"/>
  </rdf:Description>
  <rdf:Description rdfs:about="http://example.org/earth">
    <rdf:type rdfs:resource="http://example.org/planet"/>
  </rdf:Description>
  <rdf:Description rdfs:about="http://example.org/tea">
    <rdf:type rdfs:resource="http://example.org/item"/>
  </rdf:Description>
  <rdf:Description rdfs:about="http://example.org/in_search_of">
    <rdf:type rdfs:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdfs:resource="http://example.org/person"/>
    <rdfs:range rdfs:resource="http://example.org/item"/>
  </rdf:Description>
</rdf:RDF>

```

## nodes.xml

```

<?xml version="1.0"?>
<document>
  <nodes>
    <node id="1" label="Hogwarts" />
    <node id="2" label="Red London" />
    <node id="3" label="Britannia" />
    <node id="4" label="Paragaphus" />
    <node id="5" label="Ulm" />
  </nodes>
  <edges>
    <edgegroup start="5">
      <edge end="2" weight="0.4" />
      <edge end="1" weight="0.6" />
    </edgegroup>
    <edgegroup start="2">
      <edge end="5" weight="0.2" />
      <edge end="5" weight="0.01" />
    </edgegroup>
    <edgegroup start="4">
      <edge end="5" weight="1.5" />
    </edgegroup>
    <edgegroup start="1">
      <edge end="3" weight="0.8" />
      <edge end="4" weight="0.7" />
    </edgegroup>
    <edgegroup start="3">
      <edge end="5" weight="0.9" />
    </edgegroup>
  </edges>
</document>

```

**graph.xslt**

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html><body>
      <xsl:for-each select="document/nodes/node">
        <xsl:value-of select="@label" />
        <br /><ul>
          <xsl:variable name="i" select="@id" />
          <xsl:apply-templates select="/document/edges/edgegroup[@start=$i]" />
        </ul>
      </xsl:for-each>
    </body></html>
  </xsl:template>

  <xsl:template match="edgegroup">
    <xsl:for-each select="edge[@weight>0.4]">
      <xsl:variable name="j" select="@end" />
      <li>
        <xsl:value-of select="/document/nodes/node[@id=$j]/@label" />
        (<xsl:value-of select="@weight" />)
      </li>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

